(3)

(a) The test accuracy when the model is trained with full training data is: 0.82

(b)

L=1	L=2	L=3	L=4	L=5	L=6	L=7	L=8	L=9	L=10
0.565	0.645	0.715	0.715	0.765	0.74	0.74	0.805	0.8	0.82

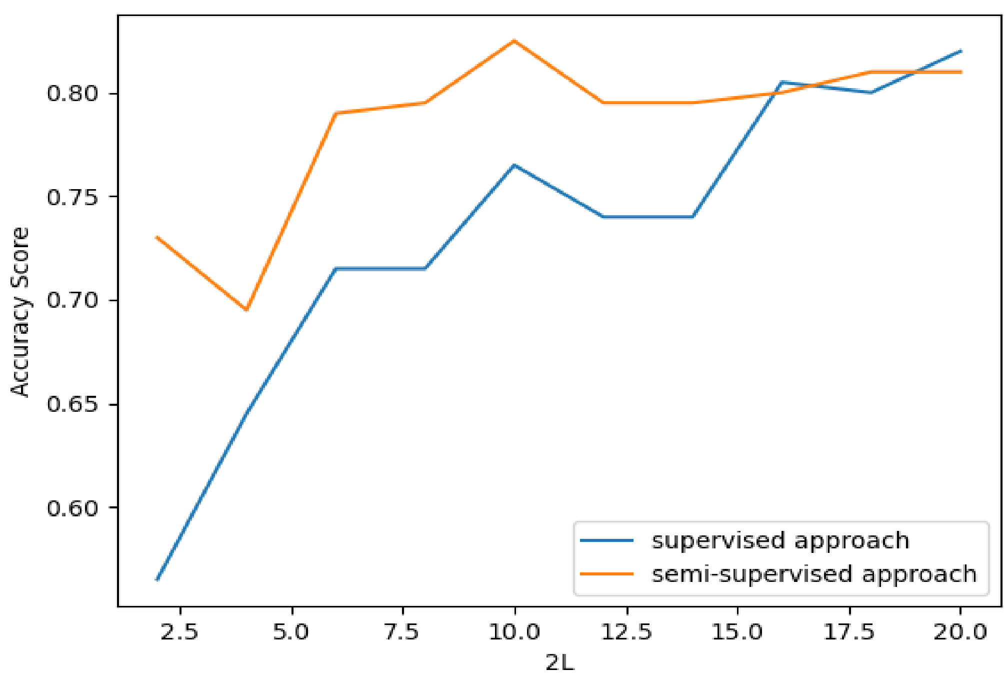
We notice that the higher the number of the training points is, the higher the accuracy we get which is expected since the models generally learns more with training.

(c)

L=1	L=2	L=3	L=4	L=5	L=6	L=7	L=8	L=9	L=10
0.73	0.695	0.79	0.795	0.825	0.795	0.795	0.8	0.81	0.81

We notice that the semi-supervised approach performed better than the supervised approach when the number of labeled data is low. i.e., using the unlabeled data for training has significantly improved the performance.

(d)



(e)

a- As expected, utilizing the unlabeled data for training has improved the prediction performance (when the number of labeled data is low) since the unlabeled data have been used to put the decision boundary of the classifier which enhanced its prediction on the test set.

In supervised approach, the few labeled data might not be descriptive to the two underlying class distributions, which means that the decision boundary will be off and there will be a lot of misclassifications. On the other hand, using the unlabeled data with the assumption that the labeled and the unlabeled distributions are consistent will result in a better position of the decision boundary.

b- when the number of labeled points is relatively high (more than 15 points) the two approaches performed equivalently because the model is not data hungry, i.e., it doesn't require lots of labeled training data in order to perform well. As observed above, we only required 20 labeled training points to reach to the best-case scenario accuracy found in part (a). So, it seems that we only require the semi-supervised approach when the labeled data are few. Otherwise, relatively higher number of labeled data used in a supervised manner will be sufficient to produce good generalization performance.

Similarly for the semi-supervised approach, the accuracy increases with the number of labeled points then it saturates, i.e., adding more labeled points doesn't further improve the performance, and that is probably because the kernel is linear and there has to be violation to its boundary since the two class distributions overlap, so the objective function probably can't be further optimized.